

Addressing blockwise erasable memories involves checking flag memory per address conversion in connection with sector write command leading to written sector to determine if block address present

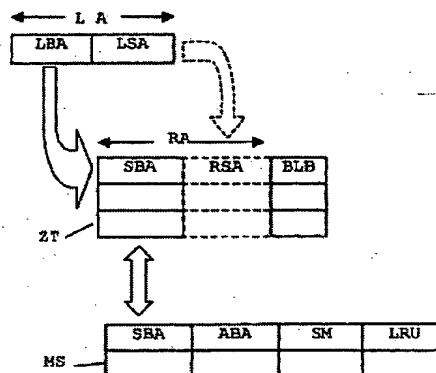
AF

Patent number: DE10227256
Publication date: 2003-12-18
Inventor: KUEHNE REINHARD (DE)
Applicant: HYPERSTONE AG (DE)
Classification:
- **international:** G06F3/06; G06F12/02; G11C8/12; G06F3/06; G06F12/02; G11C8/00; (IPC1-7): G11C16/08
- **europaean:** G06F3/06E; G06F12/02D2E2; G11C8/12
Application number: DE20021027256 20020619
Priority number(s): DE20021027256 20020619

Report a data error here

Abstract of DE10227256

The method involves checking a flag memory for each address conversion in connection with a sector write command leading to a written sector to determine if the given block address is present. If not the contents of the sectors of the relevant blocks are written to unwritten sectors of blocks assigned according to the flag memory contents. The method involves, following each address conversion in an alternative memory block address, storing it with the associated sector address assigned to the logical block address in a flag memory (MS). The flag memory is checked for each address conversion in connection with a sector write command leading to a written sector to determine if the same block address is present. If not the contents of the sectors of the relevant blocks are written to unwritten sectors of blocks assigned according to the flag memory contents



Data supplied from the esp@cenet database - Worldwide



19 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

12 Patentschrift
10 DE 102 27 256 C 1

51 Int. Cl. 7:
G 11 C 16/08

21 Aktenzeichen: 102 27 256.5-53
22 Anmeldetag: 19. 6. 2002
43 Offenlegungstag: -
45 Veröffentlichungstag
der Patenterteilung: 18. 12. 2003

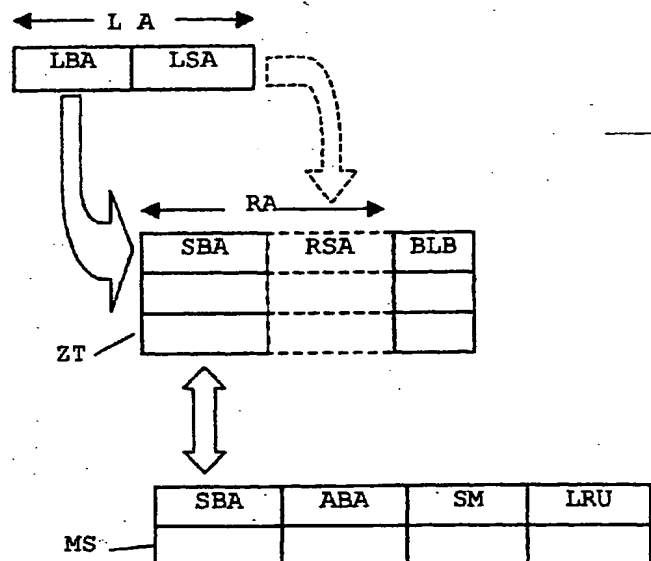
Innerhalb von 3 Monaten nach Veröffentlichung der Erteilung kann Einspruch erhoben werden

73 Patentinhaber:
Hyperstone AG, 78467 Konstanz, DE
74 Vertreter:
Boehmert & Boehmert, 33102 Paderborn

61 Zusatz in: 102 56 509.0
72 Erfinder:
Kühne, Reinhard, 78479 Reichenau, DE
56 Für die Beurteilung der Patentfähigkeit in Betracht
gezogene Druckschriften:
US 59 30 193 A

54 Verfahren zum Adressieren von blockweise löschbaren Speichern

57 Die Erfindung beschreibt ein Verfahren zum Adressieren von Speicherblöcken (SB), die in mit relativen Sektoraladressen (RSA) adressierbare einzeln beschreibbare Speichersektoren gegliedert sind, mittels einer Adressumsetzung von logischen Blockadressen (LBA) in eine der Speicherblockadressen (SBA), wobei jeweils, wenn ein Sektorschreibbefehl in einen bereits beschriebenen Sektor auszuführen ist, durch eine Adressumsetzung in einen alternativen Speicherblock (AB), dessen entsprechender Sektor gelöscht ist, in diesen geschrieben wird, dadurch gekennzeichnet, dass nach jeder Adressumsetzung in eine alternative Speicherblockadresse (ABA) diese zusammen mit der jeweils zugehörigen Sektoraladresse zugeordnet zu der logischen Blockadresse (LBA) in einem Merkspeicher (MS) gespeichert wird und bei jedem Sektorschreibbefehl, der auf einen beschriebenen Sektor führt, der Merkspeicherinhalt überprüft wird, ob die gleiche Blockadresse dort vorliegt und, wenn dies nicht der Fall ist, die Inhalte der Sektoren des betreffenden Speicherblocks (SB) in die noch unbeschriebenen Sektoren des zugeordneten Speicherblocks (ABA) geschrieben werden.



DE 102 27 256 C 1

DE 102 27 256 C 1

Beschreibung

[0001] Die Erfindung betrifft ein Verfahren zum Adressieren von einzelnen löschbaren realen mit Speicherblockadressen adressierbaren Speicherblöcken, die in mit relativen Sektoradressen adressierbare einzeln beschreibbare Speichersektoren gegliedert sind, deren gelöscht- und beschriebenen Zustände gespeichert jeweils abfragbar bereitgestellt werden, vermittels einer Adressumsetzung von logischen Blockadressen jeweils in eine der realen Speicherblockadressen, wobei jeweils, wenn ein Sektorschreibbefehl auszuführen wäre, der einen bereits beschriebenen Sektor betrifft, durch eine geänderte Adressumsetzung in eine alternative Speicherblockadresse eines Speicherblocks, dessen entsprechender Sektor gelöscht oder unbeschrieben ist, in diesen geschrieben wird und falls jedoch auch dieser alternativ adressierte Sektor bereits beschrieben war, die Adressumsetzung eine weitere Speicherblockadresse eines gelöschten Speicherblockes als eine neue alternative Speicherblockadresse bereitstellt und dessen entsprechender Sektor beschrieben wird, und im bisherigen alternativ adressierten Speicherblock die noch unbeschriebenen Sektoren mit den Inhalten der entsprechenden Sektoren des ursprünglich zugeordnet adressierten Speicherblocks beschrieben werden, wonach dieser zu einer Löschung ansteht und nach einer Löschung zu einer anderen Adressumsetzung bereitsteht.

[0002] Ein derartiges Verfahren ist aus der US 6 145 051 A bekannt. Hierbei wird in einer Adressumsetzungstabelle zu jeder logischen Blockadresse eine erste physikalische, reale Speicheradresse eines Speicherblocks gehalten und dann eine weitere reale Speicherblockadresse eingestellt, wenn ein Sektor des ersten Speicherblocks mit neuer Information beschrieben werden sollte, der bereits beschrieben war, weswegen die neu zu schreibende Information in den Ausweichblock mit der weiteren Speicherblockadresse eingeschrieben wird, weil ein sofortiger Löschvorgang relativ sehr zeitaufwendig wäre und eine vermeidbare Abnutzung des Speicherblocks verursachen würde, da die zulässige Löschhäufigkeit beschränkt ist. Außerdem sind in der Adress-Transformationstabelle Statusinformationen über die belegten/freien Sektoren und die Sektoren mit veraltetem Inhalt, die zu löschen sind, enthalten. Ein derartiges Tabellenwerk ist sehr platzaufwendig und muss wegen des häufigen und schnellen Zugriffs und wegen der ständigen Nachträge der Ausweichblockadressen und der notwendigen Umstrukturierung der Adressenzuordnung, wenn Blöcke umkopiert oder gelöscht worden sind, im Hilfsspeicher des Steuerprozessors gehalten werden, dessen Inhalt bei Stromausfall oder Abschaltung in den nichtflüchtigen Speicher gerettet werden muss.

[0003] Auch ist aus der US 5,930,193 ein Verfahren bekannt, die Speicherblöcke in Gruppen zu klassifizieren, die die bereits erfolgten Löschungen der zugeordneten Speicher in groben Bereichen angeben. Das Management dieser Gruppen erfordert weitere große Tabellen, die Platz im Merkspeicher verlangen und die beim Abschalten des Systems gesichert werden müssen.

[0004] Weiterhin ist aus der WO 00/14641 ein Verfahren zur Adressierung eines derartigen Speichers bekannt, bei dem die Zuordnung neu zu beschreibender Speicherblöcke so vorgenommen wird, dass defekte Blöcke nicht mehr benutzt werden und die einzelnen Blöcke gleichmäßig einer durch die Löschhäufigkeit verursachte Abnutzung unterliegen.

[0005] Ein Einschreiben neuer Informationen in einen bereits beschriebenen Sektor erfordert eine Verlagerung der restlichen, weiter gültigen Informationen aus den übrigen Sektoren in die eines unbeschriebenen, vorgelöschten

Blocks, dessen Adresse in den Zuordner übernommen wird. Nur wenn viele vorgelöschte Blöcke konsekutiv beschrieben werden, arbeitet dieses Verfahren zeitlich effektiv; Randomveränderungen von Speicherinhalten sind jedoch relativ aufwendig.

[0006] Es ist Aufgabe der Erfindung, ein Verfahren zu offenbaren, bei dem sowohl die Vorteile einer gleichmäßigen durch Löschvorgänge verursachten Speicherabnutzung und effektive Veränderungen von insbesondere randomadressierten Speicherinhalten durch eine Ausweichblocknutzung bei minimalem Bedarf an Zuordnungsspeicherplatz erbracht werden.

[0007] Die Lösung besteht darin, dass nach jeder, insbesondere abnutzungsoptimierenden Adressumsetzung in eine alternative Speicherblockadresse diese zusammen mit der jeweils zugehörigen Sektoradresse zugeordnet zu der logischen Blockadresse in einem Merkspeicher gespeichert wird und bei jeder Adressumsetzung in Verbindung mit einem Sektorschreibbefehl, die auf einen beschriebenen Sektor führt, der Merkspeicherinhalt daraufhin geprüft wird, ob die gleiche Blockadresse dort vorliegt und dann, wenn dies nicht der Fall ist, die Inhalte der Sektoren des betreffenden Speicherblocks in die noch unbeschriebenen Sektoren des gemäß dem Merkspeicherinhalt zugeordneten Speicherblocks geschrieben werden und dann die Speicherblockadresse des kopierten Speicherblocks in eine Adressliste von zu löschenden Speicherblöcken eingetragen wird und/oder der Block als zu löschend gekennzeichnet wird und danach dem noch anstehenden Sektorschreibbefehl eine Speicherblockadresse eines gelöschten, unbeschriebenen Speicherblocks zugeordnet und in den Merkspeicher eingetragen wird und mit dieser Speicherblockadresse der Schreibbefehl im entsprechenden Sektor ausgeführt wird.

[0008] Vorteilhafte Ausgestaltungen sind in den Unteransprüchen angegeben.

[0009] Die Adressumsetzung einer logischen in eine reale Adresse findet vorzugsweise in vorbekannter Weise so statt, dass eine gleichmäßige Löschhäufigkeit für alle Blöcke erreicht wird. Überlagert ist dem die Bereitstellung eines Ausweichblocks, wenn ein Schreibbefehl für einen bereits beschriebenen Sektor in einem originär zugeordneten Speicherblock auftritt, und in diesen Ausweichblock werden die anliegende zu schreibende Information und weitere neu zu schreibende Sektorinhalte des gleichen Blockes eingeschrieben, wozu nur die Ausweichblockadresse und ein Sektorstatus des Ausweichblockes zur logischen Blockadresse in einem kleinen Merkspeicher gehalten werden.

[0010] Sobald in einem anderen als dem originär einer logischen Adresse zugeordneten Speicherblock ein bereits beschriebener Sektor neu beschrieben werden soll, wird der Ausweichspeicherblock aus dem originären Speicherblock inhaltlich komplettiert und seine Speicheradresse einer künftig originären Adressumsetzung zugeführt, und es wird der bisherige Speicherblock mit dem veralteten Inhalt zur Löschung freigegeben. Die Bildung einer neuen Ausweichblockadresse wird hierzu mit dem die Abnutzung egalisierenden Algorithmus vorgenommen, und die Löschung des ursprünglich benutzten und frei gewordenen Speicherblocks wird als dessen Abnutzungsvorgang ausgewertet.

[0011] Da häufig mehrere Sektoren in einem logischen Block inhaltlich zu ändern sind, z. B. wenn eine Video- oder Audiodatei bearbeitet wird, wird durch das Anlegen des Ausweichblocks und der Komplettierung desselben und Freigeben des originären Speicherblocks erst bei Schreibfehlern in einen anderen Speicherblock die Zahl der Löschvorgänge erheblich reduziert und die Reaktionszeit beim Schreiben verkürzt. Die Verringerung der Löschvorgänge ergänzt die Wirkung der Adressumsetzung, die eine gleich-

mäßige Löschhäufigkeit der Speicherblöcke erbringt, so dass die Gesamtlebensdauer des Speichers erheblich vergrößert ist.

[0012] Eine vorteilhafte Weiterentwicklung besteht darin, dass dann, wenn in mehreren Dateien quasi parallel gearbeitet wird, also auf eine entsprechende Anzahl Speicherblöcke wechselweise zugegriffen wird, eine solche Anzahl von Ausweichblöcken maximal bereitgestellt wird, die vom ansteuernden Prozessor zeitlich nebeneinander bearbeitet werden kann. Demgemäß ist die Anzahl wie in üblichen Betriebssystemen eingerichtet, z. B. maximal vier. Der Merkspeicher enthält eine entsprechende Anzahl Adressenzuordnungen und Sektorzustands-Merkbits in einer Sektormaske. Zu Beginn jeder Adressermittlung werden die wenigen Adressen im Merkspeicher auf Identität geprüft und nur, wenn keine festgestellt wird, erfolgt die vorgeschriebene Adressumsetzung. Damit wird im Merkspeicher freier Platz für eine neue Adresszuordnung geschaffen und der freigegebene Speicherblock wird zum Löschen gekennzeichnet. Der Speicher- und Zeitbedarf für die Vorprüfung der Adresse mit dem Merkspeicherinhalt ist gering, und die Verringerung der Löschhäufigkeit sowie der Zeitgewinn dadurch sind ganz erheblich.

[0013] In einer vorteilhaften Ausführung wird bei der vorgeschriebenen Suche nach einem im Merkspeicher verzeichneten Speicherblock derjenige Speicherblock zum Löschen ausgewählt, der die nächstliegende logische Blockadresse zum neu zu schreibenden Block besitzt, da die Wahrscheinlichkeit hoch ist, dass dieser Block weiter mit neuen Sektoren beschrieben wird.

[0014] In einer alternativen Ausführung wird bei der vorgeschriebenen Suche nach einem im Merkspeicher verzeichneten Speicherblock derjenige Speicherblock zum Löschen ausgewählt, der am längsten nicht mehr beschrieben wurde. Zu diesem Zweck wird im Merkspeicher für jeden Ausweichblock ein Zähler geführt, der beim Schreiben in diesen Block auf einen Anfangswert initialisiert und beim Schreiben in die anderen Ausweichblöcke heruntergezählt wird. Der Block mit dem niedrigsten Zählerstand wurde damit am längsten nicht mehr beschrieben. Mit dieser Methode wird eine gleichmäßige Verteilung der Löschoperationen auf die Blöcke erreicht, was die Lebensdauer des gesamten Speichersystems erhöht.

[0015] Zweckmäßigerweise werden die Löschvorgänge in einem Hintergrundprogramm ausgeführt. Die durch die komplettierten Adressumsetzungen gekennzeichneten Blöcke werden von dem Hintergrundprogramm in eine Tabelle der zu löschenden Speicherblöcke übernommen und die Löschung durchgeführt. Nach der Löschung werden die Blöcke als frei gekennzeichnet bzw. in der Liste gelöschter Blöcke geführt, die zur weiteren Adressumsetzung genutzt werden können.

[0016] Der Zeitgewinn wird noch dadurch gesteigert, dass das Hintergrundprogramm zum Löschen und Freigeben der Speicherblöcke zu den Zeiten ausgeführt wird, wenn kein Schreib- oder Lesebefehl ansteht.

[0017] Die Zuordnung der logischen Blockadressen zu den realen Speicherblöcken erfolgt vorteilhafterweise über eine Tabelle der realen Speicherblöcke, die über die logische Blockadresse indiziert wird. Der Sektoranteil der Adressen wird bei der Indizierung nicht berücksichtigt, da die Sektoren in den logischen Blöcken genauso nummeriert sind wie in den realen Speicherblöcken. Mit diesem Tabellenaufbau wird eine kompakte Form erreicht und der Speicherplatz für die Verwaltung minimiert.

[0018] Eine kompakte Verwaltung der Speicherblöcke wird auch dadurch erreicht, dass im Merkspeicher eine Sektormaske mit der bitweisen Kennzeichnung der beschriebenen

nen bzw. freien/gelöschten Sektoren der direkt beschreibbaren Speicherblöcke geführt wird. Ebenso wird in dem Merkspeicher eine Sektormaske mit der bitweisen Kennzeichnung des Zustands der Sektoren in dem Ausweichblock aktuell geführt. Über diese Bitlisten werden dann die Speicheroperationen gesteuert.

[0019] Eine weitere Vereinfachung der Verwaltung der Speicherblöcke ergibt sich, wenn in der Zuordnungstabelle der logischen zu den realen Speicheradressen zu jedem Speicherblock ein Bit mitgeführt wird, dass den Gelöscht-Zustand des gesamten Speicherblocks angibt. In so gekennzeichnete Blöcke kann ein Sektor direkt geschrieben werden, ohne das Verfahren mit einem Ausweichblock zu benutzen. Nach dem Schreiben des ersten Sektors in einen solchen Block wird dieses Bit zurückgesetzt.

[0020] Zur schnellen Abarbeitung von Speicheroperationen ist es vorteilhaft zunächst zu prüfen, ob die betreffende Speicherblockadresse schon in den Merkspeicher aufgenommen wurde. Falls dies der Fall ist, wird anhand der Bitliste in der Sektormaske ermittelt, ob sich der aktuelle Inhalt des betreffenden Sektors im originären Speicherblock oder im Ausweichblock befindet. Bei einer Leseoperation wird direkt auf den ermittelten Sektorinhalt zugegriffen. Falls der Sektor neu geschrieben werden soll, wird, wenn sich der Sektor im Ausweichblock befindet, dieser Ausweichblock mit den Sektoren des originalen Blocks aufgefüllt und die Zuordnungstabelle so aktualisiert, dass der Ausweichblock an die Stelle des originären Speicherblocks tritt. Der bisherige Speicherblock wird zum Löschen freigegeben. Aus dem Vorrat an gelöschten Speicherblöcken wird ein neuer Ausweichblock bereitgestellt und der Inhalt des Merkspeichers entsprechend aktualisiert. Nun wird der Sektorinhalt in den neuen Ausweichblock geschrieben und die Sektormaske aktualisiert. Wurde der Speicherblock bisher nicht im Merkspeicher geführt, wird geprüft, ob der Sektor noch gelöscht ist. Falls dies der Fall ist, wird direkt in den originären Speicherblock geschrieben und das Gelöscht-Bit in der Zuordnungstabelle zurückgesetzt. Anderenfalls wird ein Datensatz im Merkspeicher gesucht, wie oben beschrieben aufgelöst und ein neuer Ausweichblock im Merkspeicher verwaltet, in den dann geschrieben wird.

[0021] In einer vorteilhaften Ausgestaltung des Verfahrens wird bei einer Schreiboperation für einen Sektor anhand der Sektormaske im Merkspeicher überprüft, ob in dem Speicherblock bisher dieser Sektor und nur eine kleine Anzahl weiterer Sektoren, z. B. maximal bis zu einem Viertel der Sektoren eines Blockes, in den Ausweichblock geschrieben wurden. In diesem Fall wird ein neuer Ausweichblock eingerichtet, in den der veränderte Sektor geschrieben und die bisher beschriebenen Sektoren aus dem alten Ausweichblock kopiert werden. Der alte Ausweichblock wird zum Löschen freigegeben. Das Kopieren der nicht veränderten Sektoren entfällt. Damit wird eine Geschwindigkeitssteigerung der Schreiboperation erreicht.

[0022] Vorteilhafterweise wird in einem nichtflüchtigen Speicherblock eine Rekonstruktionstabelle geführt, aus der sich der Zustand des Merkspeichers nach einem Stromausfall rekonstruieren lässt. Dazu werden für alle Schreib- und Löschoperationen in den Speicherblöcken entsprechende Einträge nacheinander in die Tabelle geschrieben. Durch Nachvollziehen der Tabelleneinträge im Merkspeicher beim Systemanlauf wird der Merkspeicher auf den aktuellen Stand der Schreib- und Löschoperationen gebracht. Insbesondere wird hierbei festgestellt, welche Speicherblöcke als Ausweichblöcke in Benutzung sind und wo sich aktuelle Sektoren in den Ausweichblöcken befinden.

[0023] Zweckmäßigerweise wird für die Rekonstruktionstabelle ein Speicherblock genutzt und besitzt eine definierte

Anzahl, z. B. 127, von Einträgen. Sobald die Tabelle so lang wird, dass diese Grenze erreicht wird, werden alle Speicherzuordnungen im Merkspeicher nach dem vorgeschriebenen Verfahren aufgelöst und die Datensätze damit freigemacht und der Grundzustand hergestellt. Danach wird die Rekonstruktionstabelle gelöscht und zum neuen Schreiben eingerichtet. Während der Zeit des Auflöserns und des Löscherens der Tabelle werden keine neuen Aufträge für Speicheroperationen des übergeordneten Betriebssystems angenommen.

[0024] Die Grenze für die Größe der gesamten Speicheranordnung, die mit einem Controller verwaltet werden kann, ergibt sich aus der Größe der Adresszeiger für die Speicherblöcke von 16 Bit und der Anzahl der Speicherbausteinauswahlsignale, die vom Controller bereit gestellt werden. Vorteilhaft ist es nun, mehrere Speicherbaustein-Gruppen vorzusehen, die mit jeweils gleichgearteten Sektoren und 16 Bit-Adresszeigern für die Speicherblöcke adressiert werden. Die Anzahl der Speicherbaustein-Gruppen entspricht einer Binärzahl, zum Beispiel 4. Entsprechend dieser Anzahl wird die Größe der mit einem Controller verwaltbaren Speicheranordnung vervielfacht. Die niedrigsten binären Stellen der logischen Sektoradresse werden decodiert und zur Auswahl der Speicherbaustein-Gruppe genutzt. Dieser Teil der Sektoradresse wird auch als Index für das Array der Verwaltungsstrukturen verwendet. Damit können die Verwaltungsstrukturen mit Adresszeigern, Ausweichspeicherblöcken etc., wie bisher beschrieben, weiter genutzt werden. Der Speicher stellt sich organisatorisch als eine große Speicheranordnung mit einem Mehrfachen an Sektoren dar. Durch diese Abbildung aufeinander folgender logischer Adressen auf parallele Speicherblöcke können die Speicheroperationen quasi parallel ablaufen, und damit wird ein erheblicher Geschwindigkeitsvorteil erreicht.

[0025] Falls mehrere Speicherbaustein-Gruppen eingerichtet sind, ist es vorteilhaft, das Löschen der Blöcke in den Speicherbausteinen so zu staffeln, dass parallel in die verschiedenen Speicherbausteine Löschbefehle geladen werden und dann eine bausteintypische Löschezit insgesamt abgewartet wird, bevor die Bausteine wieder abgefragt werden, ob sie bereit für weitere Speicheroperationen sind. Die Speicherbausteine zeigen das Ende der Abarbeitung eines Befehls auch durch ein Speicherende-Signal an. Falls der Controller nur einen Eingang zum Auswerten der Endezeiten besitzt, werden die Signale der Bausteine am Controller zu einem Ende-Signal zusammengeführt und ein direktes Auswerten der einzelnen Zeiten ist nicht möglich. Wenn alle Bausteine wieder bereit sind für neue Speicheroperationen, wird das Signal am Controller entsprechend gesetzt. Durch das Warten wird ein dauerndes Abfragen der Bausteine oder des Speicherende-Signals vermieden, was zu einem erhöhten Stromverbrauch führen würde.

[0026] Ebenso wie beim Löschen ist es auch beim Schreiben vorteilhaft, dass parallel in die verschiedenen Speicherbausteine Schreibbefehle geladen werden und dann eine bausteintypische Zeit abgewartet wird, bevor die Bausteine wieder, genauso wie beim Löschen, abgefragt werden, ob sie für weitere Speicheroperationen bereit sind. Neben der Stromersparnis wird auch eine höhere Geschwindigkeit des Speichersystems erreicht, da eine entsprechende Anzahl Sektoren parallel geschrieben wird.

[0027] Vorteilhaft ist es auch, wenn der Controller zum Warten auf das Ende der Schreib- oder Löschoption in den Stromspar-Modus versetzt wird und das Schreibendesignal als Interrupt zum Verlassen dieses Modus genutzt wird. Dadurch wird ein weiterer Stromspareffekt bewirkt.

[0028] Falls der Controller entsprechend viele Eingänge für die Ende-Signale der Speichergruppen besitzt, kann der Controller in den Stromsparmodus versetzt und bei dem

Schreibendesignal jeder Speichergruppe wieder aktiv werden. Dies verkürzt insgesamt die Reaktionszeit und spart beim Stromverbrauch, da das Abfragen der Bausteine ganz entfällt.

5 [0029] Die Ausgestaltung der Erfindung ist in den Figuren beispielhaft beschrieben. Es zeigt:

[0030] Fig. 1 Adresszuordnung Struktur des Merkspeichers

[0031] Fig. 2 Nutzung der Ausweichdatenblöcke

10 [0032] Fig. 3 Nutzung eines zusätzlichen Ausweichdatenblockes

[0033] Fig. 4 Ablauf der Schreiboperation

[0034] Fig. 5 Abbildung auf parallele Speichergruppen

15 [0035] Fig. 1 zeigt die Umsetzung der vom übergeordneten Betriebssystem gelieferten logischen Adresse LA in die reale Speicheradresse RA. Die logische Adresse LA wird in eine logische Blockadresse LBA und eine logische Sektoradresse LSA unterteilt. Für die Umsetzung wird eine Zuordnungstabelle ZT gehalten, die angibt welche reale Speicherblockadresse SBA welcher logischen Blockadresse LBA zugeordnet ist. Für jede Speicherblockadresse SBA ist ein Datensatz in der Tabelle vorhanden. Die logische Blockadresse LBA dient als Index in die Zuordnungstabelle ZT. Die logische Sektoradresse LSA entspricht der realen Sektoradresse RSA und braucht daher nicht in der Tabelle gespeichert werden. Zu jeder realen Adresse RA wird gegebenenfalls ein Block-Gelöscht-Bit BLB mitgeführt, welches den Löschzustand des kompletten Speicherblocks angibt. Bei Benutzung dieses Bits kann der Speicherblock schneller und einfacher verwaltet werden.

20 [0036] Weiterhin ist die Struktur des Merkspeichers MS gezeigt. Für jede parallel mögliche Speicheradresszuordnung steht ein Datensatz mit den Feldern Speicherblockadresse SBA, Ausweichblockadresse ABA, Sektormaske SM und dem Nutzungszeitähler LRU zur Verfügung. Für die hier aufgeführte Speicherblockadressen SBA wird damit angezeigt, dass für Schreiboperationen ein Ausweichblock AB eingerichtet wurde, dessen Blockadresse ABA angegeben wird. Die Sektormaske SM gibt für jeden Sektor des durch die Adresse SBA adressierten Speicherblocks SB an, ob ein veränderter Inhalt in den Ausweichblock AB geschrieben wurde. Der Nutzungszähler LRU wird bei einer Schreiboperation bei dem betroffenen Datensatz auf einen Vorgabewert initialisiert und bei allen anderen Datensätzen heruntergezählt. Der Datensatz mit dem niedrigsten LRU-Wert wurde am längsten nicht für eine Schreiboperation genutzt.

25 [0037] In Fig. 2 wird die Verwendung des Ausweichblocks AB gezeigt, der mit der Sektormaske SM verwaltet wird. Für jeden der von 0 bis 31 nummerierten Sektoren des Speicherblock SB und des Ausweichblock AB wird in der 32 Bit langen Sektormaske SM an der entsprechenden Bitposition durch den Wert 1 gekennzeichnet, dass der jeweilige Sektor sich noch unverändert im Speicherblock SB befindet. Ein Wert 0 kennzeichnet, dass der entsprechende Sektor neu in den Ausweichblock AB geschrieben wurde. Dies ist in der Figur mit N gekennzeichnet. Der Sektor im Speicherblock SB ist damit nicht mehr gültig, gekennzeichnet durch ein X. Beim Anfordern eines neuen Ausweichblock wird das Paar aufgelöst indem die gültigen Sektoren des Speicherblock SB in den Ausweichblock AB kopiert werden, gekennzeichnet durch den Pfeil AL. Anschließend enthält der Ausweichblock AB nur gültige und aktuelle Sektoren und der Speicherblock SB kann gelöscht werden. Der zugehörige Datensatz im Merkspeicher MS ist wieder frei und kann für neue Schreiboperationen genutzt werden.

30 [0038] Fig. 3 zeigt die Nutzung eines weiteren Ausweichblockes NB. Wenn nur wenige Sektoren, in diesem Beispiel

zwei, in den Ausweichblock AB geschrieben wurden und einer dieser Sektoren, in diesem Beispiel der Sektor 30, soll ein weiteres Mal verändert werden, wird der Sektor mit dem Sektorschreibbefehl SS in den neuen Ausweichblock NB geschrieben. Anschließend werden die weiteren mit einer 0 in der Sektormaske SM gekennzeichneten Sektoren, hier der Sektor 29, mit dem Kopierbefehl K in den neuen Ausweichblock NB kopiert. Damit hat der neue Ausweichblock NB die Funktion des alten Ausweichblocks AB übernommen und der Block AB kann freigegeben und gelöscht werden. Beim Auflösen dieser Kombination werden die mit 1 in der Sektormaske SM gekennzeichneten Sektoren in den Block NB kopiert, hier dargestellt über den Pfeil AL.

[0039] In Fig. 4 ist der Ablauf einer Schreiboperation für einen Sektor dargestellt. Zunächst wird geprüft, ob für den adressierten Speicherblock ein Datensatz im Merkspeicher angelegt ist. Ist dies nicht der Fall, wird durch Lesen des Speicherblocks geprüft, ob der zu schreibende Sektor noch gelöscht ist. Wenn dies der Fall ist, wird direkt in den Speicherblock geschrieben und eventuell das Blockgelöscht-Bit zurückgesetzt.

[0040] Wenn für den adressierten Speicherblock ein Datensatz im Merkspeicher angelegt ist, wird anhand der Sektormaske geprüft, ob schon ein veränderter Sektorinhalt in den Ausweichblock geschrieben wurde. Wenn dies nicht der Fall ist, wird direkt in den Ausweichblock geschrieben und die Sektormaske wird aktualisiert.

[0041] Falls der Sektor im Ausweichblock schon geschrieben wurde, wird anhand der Sektormaske überprüft, ob bisher wenige Sektoren, z. B. maximal ein Viertel der Anzahl der Sektoren eines Speicherblocks, im Ausweichblock geschrieben sind. Falls dies nicht der Fall ist, werden die geschriebenen Sektoren aus dem originären Speicherblock in den Ausweichblock kopiert und die Zuordnungstabelle so aktualisiert, dass der bisherige Ausweichblock nun an die Stelle des originären Speicherblocks tritt. Der originäre Speicherblock wird jetzt zum Löschen freigegeben. Aus dem Vorrat gelöschter Speicherblöcke wird ein neuer Ausweichblock bereitgestellt. Falls kein gelöschter Speicherblock zur Verfügung steht, wird zunächst ein freigegebener Speicherblock gelöscht. Mit diesem neuen Ausweichblock wird der Datensatz im Merkspeicher aktualisiert. Nun wird der adressierte Sektor in den neuen Ausweichblock geschrieben und die Sektormaske im Merkspeicher aktualisiert.

[0042] Falls der neu zu schreibende Sektor nur einer von wenigen geschriebenen Sektoren im Ausweichblock war, wird ein weiterer Ausweichblock bereitgestellt. Die bisher geschriebenen Sektoren aus dem alten Ausweichblock werden in den neuen Ausweichblock kopiert, der alte Ausweichblock wird freigegeben und der Merkspeicher entsprechend aktualisiert. Der neue Sektor wird nun in den Ausweichblock geschrieben und die zugehörige Sektormaske aktualisiert.

[0043] Falls dem Speicherblock bisher kein Datensatz im Merkspeicher zugeordnet war, aber der adressierte Sektor nicht gelöscht ist, wird ein Datensatz im Merkspeicher gesucht, dessen Speicherblockadresse dem adressierten Speicherblock am nächsten liegt oder am längsten nicht beschrieben wurde. Dieser Merkspeicher-Datensatz wird aufgelöst, indem die Sektoren des originären Speicherblocks in den zugehörigen Ausweichblock kopiert werden, die Zuordnungstabelle aktualisiert wird und der originäre Speicherblock freigegeben wird. Über einen neuen Ausweichblock und die Aktualisierung des Datensatzes im Merkspeicher kann nun der Sektor in den neuen Ausweichblock geschrieben werden. Anschließend wird die Sektormaske aktualisiert.

[0044] Fig. 5 zeigt die Umsetzung der logischen Adresse,

bestehend aus der logischen Blockadresse LBA und der logischen Sektoradresse LSA in die reale Speicheradresse, bestehend aus der Speicherblockadresse SBA, der realen Sektoradresse RSA und den Bits für die Gruppennummer GNx.

5 Der Buchstabe x wird hier als laufender Index benutzt. Hier sind beispielhaft 2 Bits, GN0 und GN1, gezeigt, die mittels der Decodierung DEC zur Auswahl der 4 Speichergruppen SG0-SG3 genutzt werden. Die reale Sektoradresse RSA entspricht in der Stellenanzahl der Anzahl der Sektoren in den Speicherblöcken SGn-x. Mit der realen Sektoradresse RSA werden die Sektoren innerhalb eines Speicherblocks SBN-x adressiert. Die höchstwertigen Bits der realen Speicheradresse bilden die Speicherblockadresse SBA, die zur Adressierung der Speicherblöcke in allen Speichergruppen gleichzeitig dient. Somit stellt sich ein Speicherblock mit einem Mehrfachen an Sektoren dar. Durch diese Abbildung aufeinander folgender logischen Adressen auf die Sektoren mehrerer Speicherblöcke SBN-x werden Speicheroperationen, die aufeinander folgende logische Sektoren betreffen, auf parallele Speichergruppen abgebildet und die Speicheroperationen können in den Speichergruppen SGx quasi parallel ablaufen.

[0045] Bei der Initialisierung des Speichersystems wird festgelegt, wie viele parallele Speichergruppen gebildet werden. Die maximale Zahl ist die Anzahl der eingesetzten Speicherbausteine. Der Decoder DEC wird durch den Speicher-Controller entsprechend eingestellt.

[0046] Die Speicherbausteine generieren Schreibendesignale SESx, die das Ende einer Schreib- oder Löschoperation angeben. Diese Signale werden in einer Ausführung in einer ODER-Funktion zusammengefasst und als Interrupt-Signal INT dem Speichercontroller zugeführt. Auf diese Weise kann der Controller aus dem Stromspar-Zustand aktiviert werden oder im Programm des Controllers durch Abfragen festgestellt werden, wann die Speicheroperationen abgelaufen sind.

[0047] In einer anderen Ausführung werden die Speicherehendesignale SESx direkt auf den Speichercontroller geführt und können dort abgefragt oder für einen spezialisierten Interrupt genutzt werden.

Bezugszeichenliste

	AB Ausweichblock (Zwilling)
45	ABA Ausweichblockadresse
	AL Auflösung
	BLB Blockgelöscht-Bit
	DEC Decoder für Speichergruppen
	GNx Gruppennummer x
50	INT Interrupt für Schreibende
	K Kopieren
	LA Logische Adresse
	LBA Logische Blockadresse
	LRU Zähler für Alter des Sektors
55	LSA Logische Sektoradresse
	MS Merkspeicher
	N Neuer Sektorinhalt
	NB Neuer Ausweichblock (Drilling)
	RA Reale Adresse
60	RSA Reale Sektoradresse
	RT Rekonstruktionstabelle (Logbuch)
	SB Speicherblock
	SBA Speicherblockadresse
	SBN-x Speicherblock n-x
65	SESx Schreibendesignal x
	SGx Speichergruppe x
	SM Sektormaske
	SS Sektor schreiben

WR Schreibauftrag für einen Sektor
 x veralteter Sektorinhalt
 ZT Zuordnertabelle

Patentansprüche

1. Verfahren zum Adressieren von einzelnen löschbaren realen mit Speicherblockadressen (SBA) adressierbaren Speicherblöcken (SB), die in mit relativen Sektoradressen (RSA) adressierbare einzeln beschreibbare Speichersektoren gegliedert sind, deren gelöscht- und beschriebenen Zustände gespeichert jeweils abfragbar bereitgestellt werden, mittels einer Adressumsetzung von logischen Blockadressen (LBA) jeweils in eine der Speicherblockadressen (SBA), wobei jeweils, wenn ein Sektorschreibbefehl auszuführen wäre, der einen bereits beschriebenen Sektor betrifft, durch eine geänderte Adressumsetzung in eine alternative Speicherblockadresse (ABA) eines Speicherblocks (AB), dessen entsprechender Sektor gelöscht oder unbeschrieben ist, in diesen geschrieben wird und falls jedoch auch dieser alternativ adressierte Sektor bereits beschrieben war, die Adressumsetzung eine weitere Speicherblockadresse (SBA) eines gelöschten Speicherblockes (SB) als eine neue alternative Speicherblockadresse (ABA) bereitstellt und dessen entsprechender Sektor beschrieben wird und im bisherigen alternativ adressierten Speicherblock (AB) die noch unbeschriebenen Sektoren mit den Inhalten der entsprechenden Sektoren des ursprünglich zugeordnet adressierten Speicherblocks (SB) beschrieben werden, wonach dieser zu einer Löschung ansteht und nach einer Löschung zu einer anderen Adressumsetzung bereitsteht, **dadurch gekennzeichnet**, dass nach jeder, insbesondere abnutzungsoptimierenden Adressumsetzung in eine alternative Speicherblockadresse (ABA) diese zusammen mit der jeweils zugehörigen Sektoradresse zugeordnet zu der logischen Blockadresse (LBA) in einem Merkspeicher (MS) gespeichert wird und bei jeder Adressumsetzung in Verbindung mit einem Sektorschreibbefehl, die auf einen beschriebenen Sektor führt, der Merkspeicherinhalt daraufhin geprüft wird, ob die gleiche Blockadresse dort vorliegt und dann, wenn dies nicht der Fall ist, die Inhalte der Sektoren des betreffenden Speicherblocks (SB) in die noch unbeschriebenen Sektoren des gemäß dem Merkspeicherinhalt zugeordneten Speicherblocks (ABA) geschrieben werden und dann die Speicherblockadresse (SBA) des kopierten Speicherblocks in eine Adressliste von zu löschenden Speicherblöcken eingetragen wird und/oder der Block als zu löschend gekennzeichnet wird und danach dem noch anstehenden Sektorschreibbefehl eine Speicherblockadresse (ABA) eines gelöschten, unbeschriebenen Speicherblocks zugeordnet und in den Merkspeicher eingetragen wird und mit dieser Speicherblockadresse (ABA) der Schreibbefehl im entsprechenden Sektor ausgeführt wird.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass der Merkspeicher (MS) nur so viele Speicheradresszuordnungen enthält, wie Speicherdaten parallel zu bearbeiten sind, und immer dann, wenn mit einem Sektorschreibbefehl eine Adressumsetzung auf einen beschriebenen Sektor führt, sämtliche Zuordnungen im Merkspeicher auf eine Identität der Blockadressen (SBA) überprüft werden und, wenn keine identische Blockadresse (SBA) vorliegt, die Sektoren eines Blocks (SB) in die noch unbeschriebenen Sektoren des zugeordneten Speicherblocks (AB) übertragen werden

und dann die Speicherblockadresse (SBA) des kopierten Blocks (SB) in eine Adressliste von zu löschenden Speicherblöcken eingetragen wird und/oder der Block als zu löschend gekennzeichnet wird und danach dem noch anstehenden Sektorschreibbefehl eine Speicherblockadresse (ABA) eines gelöschten, unbeschriebenen Speicherblocks (AB) zugeordnet und in den Merkspeicher (MS) eingetragen wird und mit dieser Blockadresse (ABA) der Schreibbefehl im entsprechenden Sektor ausgeführt wird.

3. Verfahren nach Anspruch 2, dadurch gekennzeichnet, dass für eine neue Adressumsetzung der im Merkspeicher (MS) verzeichnete Block (SB) freigemacht wird, der die nächstliegende logische Blockadresse (LBA) besitzt.

4. Verfahren nach Anspruch 2, dadurch gekennzeichnet, dass für eine neue Adressumsetzung der im Merkspeicher (MS) verzeichnete Block (SB) freigemacht wird, bei dem ein Schreibzähler (LRU) angibt, dass er am längsten nicht beschrieben wurde.

5. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass in einem Hintergrundprogramm die Adressliste der zu löschenden Speicherblöcke abgearbeitet und die Löschungen durchgeführt werden und die Adressen der gelöschten Speicherblöcke in eine Liste gelöschter Speicherblöcke eingetragen werden und/oder diese Blöcke als gelöschte Blöcke gekennzeichnet werden.

6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, dass das Hintergrundprogramm die Löschungen jeweils dann durchführt, wenn kein Schreib- oder Lesebefehl ansteht.

7. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass die Speicheradresszuordnung über eine mit der logischen Blockadresse zu adressierenden Zuordnertabelle (ZT) erfolgt.

8. Verfahren nach Anspruch 7, dadurch gekennzeichnet, dass in dem Merkspeicher (MS) zu jeder Blockadresse in einer den einzelnen Sektoren bitweise zugeordneten Sektormaske (SM) die jeweils beschriebenen Sektoren aktuell geführt werden.

9. Verfahren nach Anspruch 8, dadurch gekennzeichnet, dass in der Zuordnertabelle (ZT) zu jeder Blockadresse ein Bit (BLB) mitgeführt wird, dass den "Gelöscht-Zustand" des Speicherblocks angibt.

10. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass zu Beginn jeder Schreiboperation der Merkspeicherinhalt daraufhin überprüft wird, ob die logische Block- und Sektoradresse darin enthalten sind und in welchem der Speicherblöcke (SB) ein jeweils aktueller Sektorinhalt vorliegt, der durch eine zu schreibende Information in den freien Sektor eines jeweils zugeordneten oder noch zuzuordnenden Speicherblockes (AB) ersetzt wird.

11. Verfahren nach Anspruch 10, dadurch gekennzeichnet, dass, falls in einem Speicherblock (AB) bisher wenige Sektoren verändert wurden und einer dieser Sektoren anschließend nochmals verändert werden soll, dieser Sektor in einen neu zugeordneten Speicherblock (NB) geschrieben wird und die anderen veränderten Sektoren in diesen neuen Speicherblock (NB) aus dem bisherigen Speicherblock (AB) kopiert werden, und dieser neue Speicherblock (NB) an die Stelle des bisher zugeordneten Speicherblocks (AB) tritt.

12. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass in einem Speicherblock eine Rekonstruktionstabelle (RT) für jede Schreib- oder Löschoperation geführt wird, aus der

sich der Zustand des Merkspeichers (MS) nach einem Stromausfall beim Wiederanlauf des Systems rekonstruieren lässt.

13. Verfahren nach Anspruch 12, dadurch gekennzeichnet, dass bei Erreichen eines vordefinierten Füllstandes der Rekonstruktionstabelle (RT) ein definierter Grundzustand des Merkspeichers (MS) und der Rekonstruktionstabelle (RT) herbeigeführt wird.

14. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass in einer Speicheranordnung mehrerer unabhängig voneinander anwählbarer und löschbarer Speicherbaustein-Gruppen (SGx) die logischen Speicheradressen (LA) den Speicherblockadressen (SBA) und den Sektoradressen (LSA) der Speicherbaustein-Gruppen (SGx) derart zugeordnet werden, dass die Anzahl der Speicherbaustein-Gruppen einer Binärzahl entspricht und dass die niedrigsten binären Stellen der logischen Sektoradresse (LSA) jeweils decodiert der Speicherbaustein-Gruppenanwahl dienen und die nächst höherwertigen binären Stellen der logischen Sektoradresse (LSA) und der höherwertig anschließenden logischen Blockadresse (LBA) als die reale Sektoradresse (RSA) aller Speicherbausteine in den Speicherbaustein-Gruppen (SGx) parallel dienen und die höchstwertigen Stellen der logischen Blockadressen (LBA) in dem Adressumsetzer der realen Speicherblockadresse (SBA) jeweils parallel allen Speicherbaustein-Gruppen (SGx) zugeordnet werden, wodurch die zugehörigen Speicherblöcke (SBn-x) organisatorisch jeweils als ein einziger Speicherblock mit einem entsprechenden Mehrfachen an Sektoren behandelt werden.

15. Verfahren nach Anspruch 14, dadurch gekennzeichnet, dass das Löschen der Blöcke der einzelnen Speicherbausteine mit einer zeitlich so gestaffelten Blockadressierung und Speicherbausteinanwahl erfolgt, dass die jeweiligen Blockadressen (SBA) und ein Löschkommando fortlaufend in die einzelnen Bausteine übernommen werden und anschließend eine jeweils bausteintypische Löschezit programmgemäß insgesamt abgewartet wird bis eine weitere Speicheroperation ausgeführt wird.

16. Verfahren nach Anspruch 14, dadurch gekennzeichnet, dass das Schreiben der Blöcke der einzelnen Speicherbausteine mit einer so zeitlich gestaffelten Block- und Sektoradressierung sowie Speicherbausteinanwahl erfolgt, dass jeweils eine Übernahme zu schreibender Information, der Adressen der Speicherbausteinanwahl und eines Schreibkommandos fortlaufend in die einzelnen Bausteine erfolgt und anschließend eine bausteintypische Schreibzeit abgewartet wird, bis weitere Speicheroperationen erfolgen.

17. Verfahren nach Anspruch 14 oder 16, dadurch gekennzeichnet, dass nach einer Schreib- oder Löschoperation der Controller in den "Powerdown-Modus" versetzt wird und dieser Modus durch einen Interrupt (INT) nach dem Ende der Operation in allen Speicherbausteingruppen (SGx) wieder verlassen wird.

18. Verfahren nach Anspruch 17, dadurch gekennzeichnet, dass für jede Speicherbausteingruppe (SGx) ein eigenes Interrupt-Signal (SESx) zur Verfügung steht.

- Leerseite -

Fig. 1

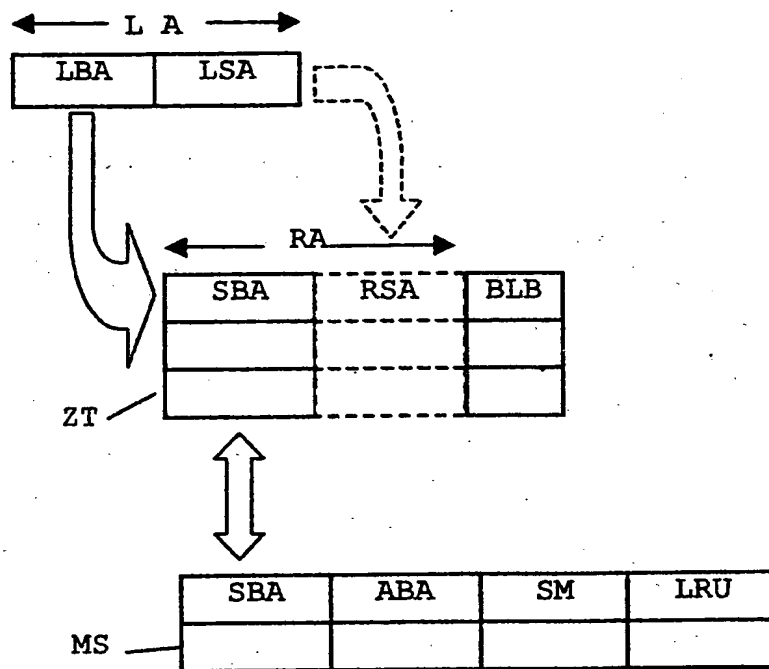


Fig. 2

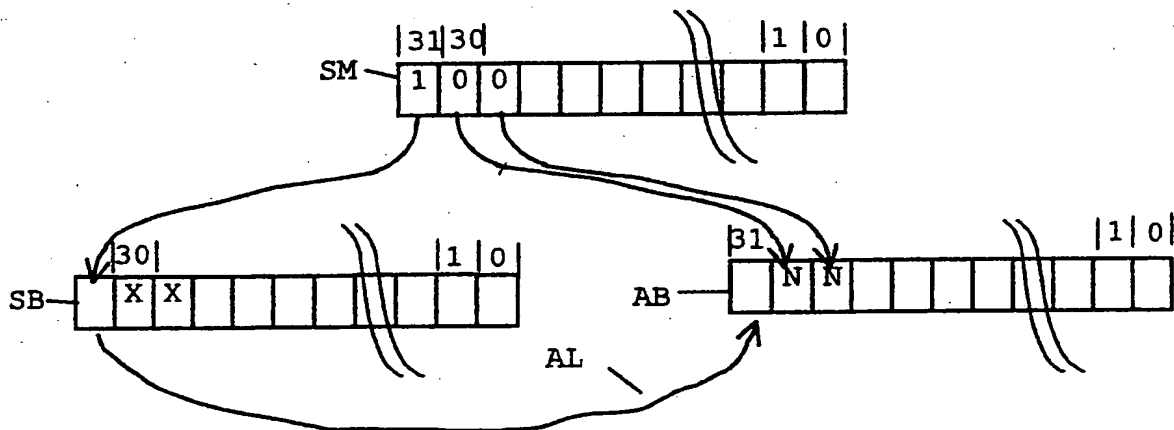


Fig. 3

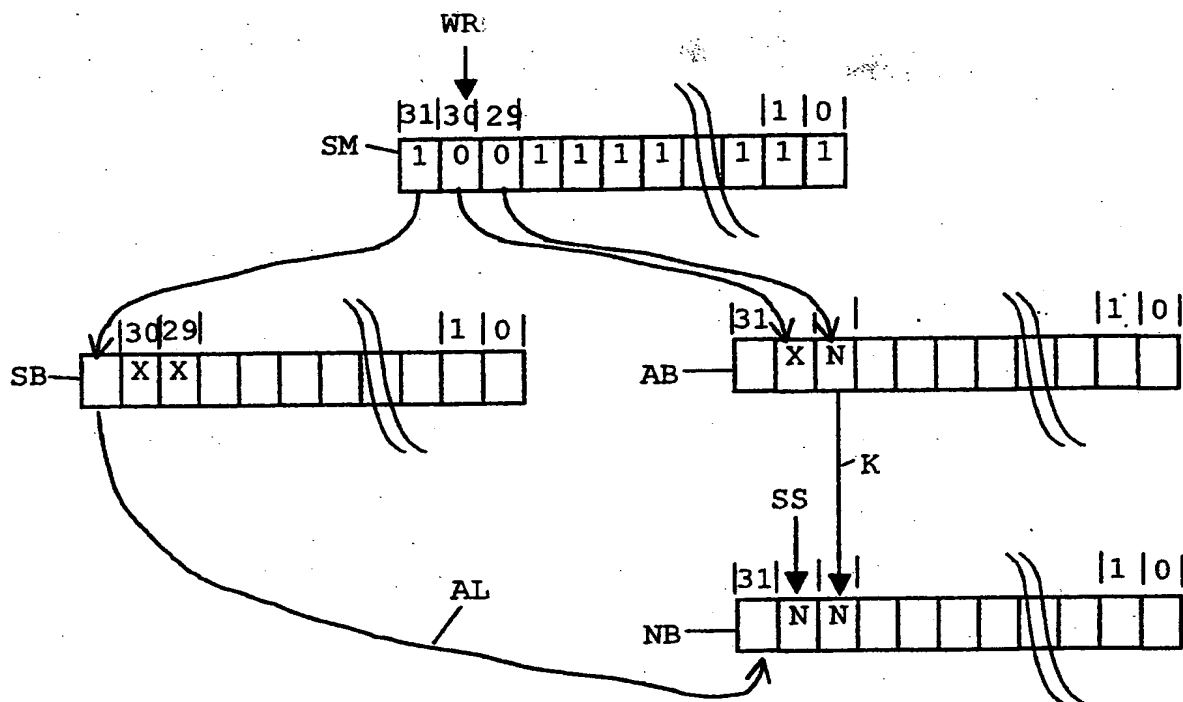


Fig. 4

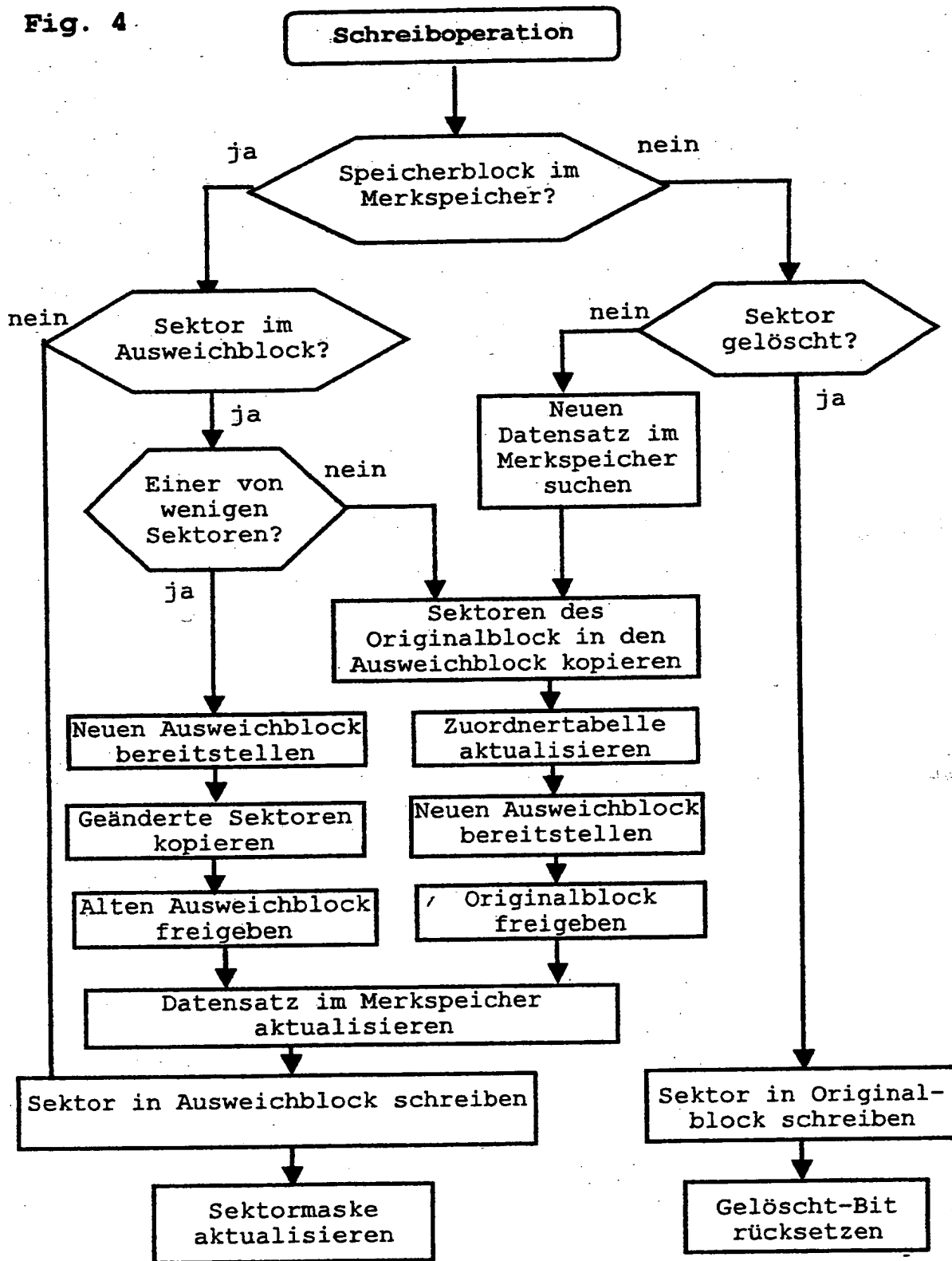


Fig. 5

